

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	IS&R	L15	10	((("5761667") or ("5881379") or ("5933820") or ("5991761") or ("6092086"))).PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:01
2	BRS	L16	0	15 and "management block"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:02
3	BRS	L17	5	15 and block	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:03
4	BRS	L18	1082	"management block"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:03
5	BRS	L19	25	18 with synchroni\$8	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:13
6	BRS	L20	2	19 with database	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:48
7	BRS	L21	6300	description near4 database	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:49
8	BRS	L22	1	18 same 21	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:49
9	BRS	L24	2	23 and synchroni\$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:50
10	BRS	L23	7	18 and 21	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:58
11	BRS	L25	0	"databae description"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:58
12	BRS	L26	327	"database description"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:58
13	BRS	L27	1	26 and 18	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/04/18 15:58

	Comments	Error Definition	Errors
1			0
2			0
3			0
4			0
5			0
6			0
7			0
8			0
9			0
10			0
11			0
12			0
13			0

PGPUB-DOCUMENT-NUMBER: 20030046294

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030046294 A1

TITLE: Symmetrical database data set allocation

PUBLICATION-DATE: March 6, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
RULE-47			
Heronimus, Scott D.	Sugar Land	TX	US

US-CL-CURRENT: 707/100

ABSTRACT:

Techniques to allocate storage space for database data sets based on the database's internal/logical boundaries is described. Metadata describing the structure and logical size requirements for various database sections are interrogated and used to guide the allocation of physical storage space on one or more storage devices. Data set extents allocated in symmetry with database internal boundaries can improve the physical database's input-output performance and storage device utilization.

----- KWIC -----

Summary of Invention Paragraph - BSTX (6):

[0005] Databases are defined in IMS by the Database Description control block ("DBD"). The DBD specifies the required geometry, attributes and access method to be used for the underlying database data set. Access methods are typically Virtual Storage Access Method ("VSAM") or Overflow Sequential Access Method ("OSAM"). A data set is usually established by invoking operating system services to allocate storage space on Direct Access Storage Devices ("DASD"). The size of a data set is defined by the amount of primary, and optionally secondary, space quantities. In the past, such space quantities have generally been arbitrary, static values that have no relationship to the associated DBD apart from the necessity of meeting the minimum required size of the database.

Summary of Invention Paragraph - BSTX (10):

[0009] Therefore, the prior art methods of determining the size and allocation of database data sets can result in database I/O constraints and DASD usage inefficiencies. Techniques in accordance with the invention solve these problems by providing an improved system and method for allocating database data sets in symmetry with the internal logical dimensions or boundaries associated with the database description control blocks. Furthermore, the present invention provides a dynamic method of adjusting the size of a physical data set extent to match the internal dimensions of a logical data structure.

Detail Description Paragraph - DETX (2):

[0019] In accordance with one aspect of the invention, logical boundaries identified in Database Description ("DBD") blocks are used to guide the physical allocation of database data sets. One embodiment of the invention is

implemented as a software utility that is executed before running an application program (e.g. a program that uses the underlying database). For ease of description, the following embodiments are described in terms of the following three IMS databases: Data Entry Database ("DEDB"); Hierarchical Direct Access Method ("HDAM"); and Hierarchical Indexed Direct Access Method ("HIDAM"). All of which typically exercised from within an associated mainframe operating systems such as OS/390 and z/OS provided by IBM corporation of Armonk, N.Y. Each of these database types has its own peculiar characteristics which will be described separately as they relate to symmetrical database data set allocation. However, as one of ordinary skill in the art will appreciate, the principles of the present invention are not limited in application to the foregoing three IMS databases, and specifically, the inventive techniques are applicable to other databases that include an internal logical structure such as relational databases, for example, DB2 by the IBM Corporation or Oracle databases by the Oracle corporation of Redwood City, Calif.

Detail Description Paragraph - DETX (12):

[0029] After initialization by a user such as a DBA (block 405), the database's Data Management Block ("DMB") is interrogated (block 410). The DMB is a structure defined by IMS that describes a database's logical structure and its physical attribute. It will be recognized by those of ordinary skill that databases other than IMS have similar structures to provide such information. Following DMB interrogation, a primary data set allocation quantity is calculated from information contained in the DMB (block 415). One of ordinary skill in the art will recognize that in an IMS environment, the DMB is a processed version of the DBD, incorporating substantially the same information as the DBD. The calculated primary data set allocation quantity may be used for the initial database data set allocation (block 420). For example, if the database is a DEDB or HDAM database, the primary quantity used will typically be based on the size of the RAA section (see FIGS. 1 and 2). If, on the other hand, the database is a HIDAM database, the calculated primary quantity will generally be based on the size of a first bit map range in the HIDAM data component (see FIG. 3). Next, the allocated data set's high allocation address is checked against the required high allocation quantity in the DMB. If the data set is sufficiently large and no additional storage is needed (the "YES" prong of diamond 425), method 400 is complete (block 430). If the allocated data set is not sufficiently large (the "NO" prong of diamond 425), a secondary allocation quantity may be calculated (block 435) based on, for example, the size of the OVFL section if the database is a HDAM or HIDAM database or the addressable range of a subsequent bit map if the database is a HIDAM database. Once calculated, storage is allocated to create a secondary quantity that is used to extend the database's data set's size (block 440). The acts of blocks 425 through 440 are repeated as necessary to allocate the needed storage.

Claims Text - CLTX (4):

3. The method of claim 1, wherein the act of obtaining internal boundary information comprises obtaining a database description control block information associated with the database data set.

Claims Text - CLTX (5):

4. The method of claim 3, wherein the act of obtaining internal boundary information comprises scanning a database management block associated with the database data set.

Claims Text - CLTX (19):

18. The method of claim 17, wherein the act of scanning comprises scanning database description control block information associated with an Information Management System database.

Claims Text - CLTX (20):

19. The method of claim 18, wherein the act of scanning database description control block information comprises scanning a database management block.

Claims Text - CLTX (24):

23. The method of claim 22, further comprising: comparing the size of the acquired storage space with the total space required by the database data set as indicated by the database description control block information; and acquiring additional storage space approximately equal to the difference between the acquired storage space and the total space required by the database data set.

Claims Text - CLTX (31):

30. The program storage device of claim 29, wherein the instructions to obtain internal boundary information comprise instructions to obtain a database description control block information associated with the database data set.

Claims Text - CLTX (32):

31. The program storage device of claim 30, wherein the instructions to obtain a database description control block information comprise instructions to obtain a database management block associated with the database data set.

Claims Text - CLTX (43):

42. The program storage device of claim 41, wherein the instructions to scan comprise instructions to scan a database description control block associated with an Information Management System database.